

MODÜL 5

LİSTELER VE ÖZELLİKLERİ



Şekil 5.1: Bölümle ilgili örnek uygulamalara karekoddan ulaşabilirsiniz.

Programlamada bir değişken üzerinde sadece bir değer tutulabilmektedir. Listeler ise bir değişkenin altında birden fazla değer tutulabilmesine yarar. Listeler, içinde **farklı türlerden verileri barındırabilen** taşıyıcılar olarak adlandırılmaktadır. Listeler Python'daki veri tiplerinden biridir. Listeler sıralı olarak kaydedilebilen veri yapılarıdır. Verilere döngü gibi yapılarla sıralı olarak erişmek istenildiğinde bize büyük avantaj sağlayıp, iki köşeli parantez arasında tanımlanırlar. Listelerde sözlüklerden farklı olarak dilimlenebilir ve elemanları sonradan değiştirilebilir. Aşağıda görüldüğü üzere boş liste ve int, string ve float verilerini bulunduran listeler tanımlanmıştır.

MODÜL 5

```
liste= []          #boş list
liste=list()
liste1=[1,2,3,4,5,6]
liste2=['a','b','c']
liste3=[0.5,1.7,67.89,3.14]
liste4=['ali','veli','yılmaz','hayri']
liste5 = list(ifade for degisken in sequence) # Hesaplanan bir liste
```

5.1. Liste Veri Tipleri

Bir listede her veri tipinden eleman saklanabilir. Bu anlamda sıralı bir diziye benzemektedir.

Örnek

1

```
liste=[1,2,'ali',0.25]
print(liste)
[1, 2, 'ali', 0.25]
```

Örnek 1’de int, string ve float gibi farklı veri tiplerini içerisinde barındıran 4 elemanlı bir listedir.

Karakter dizilerin indis değerleri değiştirilmek istendiğinde aşağıdaki gibi hata mesajı alınır.

```
meyve="erik"
meyve = 5 +meyve[0:]
print(meyve)
Traceback (most recent call last):
File "<ipython-input-9-e4575713155c>", line 2, in <module>
meyve = 5 +meyve[0:]
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

Örnek

2

```

meyve = "erik"
meyve = "ayva-" +meyve[0:]
print(meyve)
liste=['a','b']
liste[1]=2
print(liste)
ayva-erik
['a', 2]

```

Örnek 2’de karakter dizisi ile liste kullanımı beraber verilmiştir. Karakter dizisi string ama liste 1.indis numarasındaki öge ‘b’ iken sonra int veri tipinde 2 olarak değiştirilmiştir.

Listeyi oluşturmak için önce bir değişken adı verilir, daha sonra bu değişkene çeşitli değerler atanır. Atamalar yapılarak liste oluşturulur. Bir liste oluşturulup bir değişkene atandığı zaman aslında tüm atamalarda olduğu gibi liste nesnesinin adresi değişkene atanmaktadır.

Örnek 3’te oluşturulan liste içine herhangi bir eleman eklenmediği için bu liste içi boş bir listedir. Çıktı olarak boş liste vermektedir.

Örnek

3

```

liste=[ ] #veya liste=list()
print(liste)
[ ]

```

5.2. Liste Kavramı ve İndis Değerleri

Öncelikle liste üzerinden veri okuyabilmek için hangi indis elemanının okunmak istendiği doğru bir şekilde belirtilmelidir. Örneğin Harfler isminde bir liste olduğunu düşünün. Listelerde ilk eleman her zaman 0. indistir. Listenin ilk elemanına erişmek veya yazmak istendiğinde Harfler[0] yazılması gerekmektedir. Diğer elemanları için de sırasıyla Harfler[1], Harfler[2] şeklinde yazılması gerekmektedir. Buradaki bir diğer önemli nokta da listenin ilk elemanı 0. indisten başladığı için son elemanı da liste uzunluğun bir eksisidir.

MODÜL 5

Tablo 1. Listelerde indis numarasına göre veri sıralaması

Başlangıç			Bitiş(Uzunluk-1)
Harfler[0]	Harfler[1]	Harfler[2]	...

Örnek

4

```
renkler = ['Red', 'Orange', 'Yellow', 'Green', 'Blue', 'Indigo', 'Violet']
print('listeyi ekrana yazdırıyoruz')
print(renkler)
listeyi ekrana yazdırıyoruz
['black', 'Orange', 'Yellow', 'Green', 'Blue', 'Indigo', 'Violet']
```

Örnek 4’te görüldüğü üzere renkler listesindeki tüm öğeleri listelenmiştir.

Python’da listeler üzerinde değişik türden veriler bir arada tutulabilir. Python’ı güçlü kılan özelliklerden biri olan listelerde, her bir eleman bir indis (indis) numarasına sahiptir. Bir listenin başlangıç indisi 0 (sıfır)’dır.

Tablo 2. İndis numarasına göre liste örneği

İndis numarası	Liste[0]	Liste[1]	Liste[2]	Liste[3]
Öge sıralaması	Birinci öge	İkinci öge	Üçüncü öge	Dördüncü öge
Liste =	‘ali’,	5,	3.14,	‘ayşe’

Başlangıç elemanları belli olan bir listenin tanımlanması ve yazdırılması:

Örnek

5

```
#liste1 adında bir liste tanımladık. İçine verileri girdik
#bu verileri ekrana yazdırdık
liste1 = ['a','b','c','d','e','f']
print(liste1)
['a', 'b', 'c', 'd', 'e', 'f']
```

Örnek 5'te liste1 adında liste tanımlandı ve char(karakter) olarak öğeleri bulunmaktadır. İndis numarasına bakıldığı zaman; aşağıdaki şekilde öğeler vardır.

```
liste1[0] ='a'      liste1[1] ='b'      liste1[2] ='c'      liste[3] ='d'
liste1[4] ='e'      liste1[5] ='f'
```

Listeler hem string hem de sayı türünde veriler barındırabilir. Örnek 6 incelendiğinde 7 elemanlı bir liste tanımlıdır. 6 ve 7. elemanları sayı diğerleri metin olan bir listedir.

Örnek**6**

```
liste2=['python','geleceği','olan','bir','dil',500,0.567]
print(liste2)
['python', 'geleceği', 'olan', 'bir', 'dil', 500,0.567]
```

Örnek 6'da sayı, float ve string öğelerine sahip liste ekrana yazdırılmıştır.

5.3. Liste Elemanlarına Erişim

Listedeki öğelere ulaşmak için bir değer, listenin indis numarasına göre atanıp çağrılabilir. Liste oluştururken içine herhangi bir değer girilmeden de oluşturulup, sonradan değer ataması yapılabilir ya da herhangi bir indis numarasından başlanıp belirlenen indis numarasına kadar olan öğelere ulaşılır. Aşağıdaki örneklerde görüldüğü gibi hem indis numarasından hem de başlangıcı ve bitişi belli olan indis numarası aralıklarına kadar ifadelerle erişilmiştir. Listenin belirli aralıktaki öğelerini alma işlemine dilimleme denir. Liste dilimlerken adımlama da başlangıç indisi alır ama bitiş indisi almaz.

Kullanımı:

```
eleman = liste[indis]
dilim = liste[baslangic:bitis]
```

Liste, elemanlarına erişim için tamamını veya indis numarasına göre çağırılmaktadır.

MODÜL 5

Örnek

7

```
liste=["birinici veri","ikinci veri","üçüncü veri ","dördüncü veri","beşinci veri"]
#beş elemenlı listenin ilk verisi
print(liste[0])
#beş elemenlı listenin son verisi
print(liste[4])
birinici veri
beşinci veri
```

Örnek 7’de listenin indis numaraları yazılarak, başlangıç ve bitiş öğelerine ulaşılmıştır.

Örnek

8

```
liste=["ayva","armut","kiraz","vişne"]
print(liste)
print(liste[1])
['ayva', 'armut', 'kiraz', 'vişne']
armut
```

Örnek 8’de ise listenin tüm öğeleri ve 1. indis numarasındaki öge olan “armut” değeri listelenmiştir.

Liste içindeki öğeleri kontrol edilebilir. Eşya adında bir liste var. Liste tanımlanır, daha sonra if karar yapısı ile içindeki öğeler de “perde” olup olmadığı kontrol edilir.

Örnek

9

```
esya = ["ayna", "televizyon", "perde"]
if("perde" in esya):
    print("Bu değer listede var.")
else:
    print("Bu değer listede yok")
Bu değer listede var.
```

Aşağıdaki örnekte 5 elemanlı bir liste tanımlanmıştır. İndis numaralarını kullanarak, listenin 1. indisten başlayarak, 3. indise kadar öge listesine aktarıp listeleme işlemi yapılmıştır.

```
liste = [1,2,3,4,5]
öge = listem[1:3]
print(öge)
[2, 3]
```

Örnek**10**

```
listem=[10,20,30,40,50]
eleman = listem[3]
print(eleman)
40
```

Örnek 10’da listem adında bir liste tanımlanarak 5 elemanlı ifade girilmiştir. Listem adlı liste için 3. indis numarasına ait ifade eleman adlı değişkene atama işlemi yapılmıştır. Eleman adlı değişken de ekrana yazdırılarak 40 sonucu elde edilmiştir.

Örnek**11**

```
listem = [10,20,30,40,50]
eleman = listem[1:3]
print(eleman)
[20, 30]
```

Örnek 11’de listem adlı listede eleman adlı değişkene sadece 1 ve 2. indis numaralarının dâhil edildiği 3. indis numarasının dâhil edilmediği atama işlemi yapılmıştır. Eleman adlı değişken de ekrana yazdırılarak 20 ve 30 değerleri listelenmiştir.

MODÜL 5

Örnek

12

```
liste = [1,2,3,4,5,6,7,8,9,10]
print(liste)
# 1. eleman
print(liste[0])
# 6. eleman
print(liste[5])
# Baştan 5. indekse kadar (dahil değil)
print(liste[:5])
# 1.indisten 5.indise kadar
print(liste[1:7])
print(liste[5:])
print(liste[::-2])
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
1
7
[1, 2, 3, 4, 5]
[2, 3, 4, 5, 6, 7]
[6, 7, 8, 9, 10]
[1, 3, 5, 7, 9]
```

Örnek 12’de, 10 elemanlı bir liste tanımlanmış ve indis numaraları kullanılarak ekrana listeleme işlemi yapılmıştır.

Listelerde **negatif indisler** de kullanılabilir. Negatif indis numarası listenin sonuncu elemanından başlayarak sayıldığında (sondan başa) sıra numarasını verir. Örnek 13’te listenin sonuncu elemanına liste[-1] olarak, sondan ikinciye ise [-2] olarak ulaşılır. Güle güle elemanı liste de -1.indis olarak görülmektedir. Merhaba ise geriye doğru sayıldığında -4. indis’tir.

Örnek

13

```
liste = [ "merhaba", "dünya", "merhaba", "güle güle" ]
print (liste [- 1 ]) #son ögeyi listeler
print( liste [- 3 ]) #sondan üçüncü ögeyi listeler
print(liste [- 4 ]) #sondan dördüncü ögeyi listeler
print(liste[::-1]) #sondan başa doğru listeleme yapmak için kullanılır
güle güle
dünya
merhaba
['güle güle', 'merhaba', 'dünya', 'merhaba']
```

5.4. Temel Liste Metotları

Metotlar listelerin işlevlerine erişilmesini sağlar. Listenin metotları için `dir()` fonksiyonunu kullanarak tüm metotlar görülebilir. Bu metotlar yardımıyla listeler ekleme, çıkarma, arama, sıralama vb. birçok işlemin kolaylıkla yapılabilmesini sağlamaktadır.

Tablo 3. Temel liste metotları

Sıra No	Metot Adı	Görevi
1	'append'	Listeye yeni eleman ekleme işlemini yapar. Bu metot ile listeye sadece bir eleman eklenebilir ve eklenen eleman listenin sonunda yer alır.
2	'clear'	Listeyi değil içindeki tüm ifadeleri silmeye yarar.
3	'copy'	Listeden listeye kopyalama işlevine yaramaktadır.
4	'count'	Listenin içinde sorgulanan elemandan kaç adet olduğunu bulmamızı sağlar.
5	'extend'	Listeler arası genişletme işlevini görür.
6	'indis'	Listedeki elemanları almamızı sağlar.
7	'insert'	Listenin istenilen indis numarasına eleman eklenebilir.
8	'pop'	Listedeki elemanın indisi ile silme işlemi yapar. indis belirtmediğinizde ise varsayılan olarak listenin son elemanını siler. Ayrıca bu metot silinen elemanı ekrana yazmaktadır.
9	'remove'	Listede istenilen elemanın değerini yazarak silme işlemi yapar.
10	'sort'	Listenin elemanlarını alfabetik olarak sıralar.
11	'reverse'	Bu metot sort metodunun aksine listedeki elemanları ters alfabetik olarak sıralar.
12	'del'	Liste içerisinde bir elemanı silmek için kullanılır. Silme işlemi indis numarasına göre yapılmaktadır.

5.4.1. 'append' kullanımı

Örnek

14

```
takimlar=["gs","fb","bjk"]
takimlar.append("ts")
print(takimlar)
['gs', 'fb', 'bjk', 'ts']
```

Örnek 14'te takimlar listesine "ts" ögesi eklenmiş ve son indis numarasında yer almıştır.

5.4.2. 'insert' kullanımı

Örnek

15

```
sebzeler=["lahana","marul","pırasa","ıspanak","fasulye"]
sebzeler.insert(2,"patlıcan")
print(sebzeler)
['lahana', 'marul', 'patlıcan', 'pırasa', 'ıspanak', 'fasulye']
```

Örnek 15'te insert metodu kullanılarak 2. indis numarasına "patlıcan" ögesi eklenerek, listeleme işlemi yapılmıştır.

5.4.3. 'copy' kullanımı

Örnek

16

```
iller1=["konya","karaman","kocaeli","kayseri","kahramanmaraş"]
iller2=[]
iller2 = iller1.copy()
print(iller2)
['konya', 'karaman', 'kocaeli', 'kayseri', 'kahramanmaraş']
```

Örnek 16'da iller1 listesi copy metodu ile iller2 listesine aktarılmıştır.

MODÜL 5

5.4.4. 'count' kullanımı

Örnek 17

```
takimlar = ['GS', 'FB', 'BJK', 'TS']  
print(takimlar.count('FB'))  
1
```

Örnek 17’de takimlar listesinde ‘FB’ ögesinin kaç adet olduğu count metodu bulunmuştur.

5.4.5. 'extend' kullanımı

Örnek 18

```
kus1=["bildircin","papağan","kartal","akbaba","şahin"]  
kus2=["baykuş","muhabbet"]  
kus1.extend(kus2)  
print(kus1)  
['bildircin', 'papağan', 'kartal', 'akbaba', 'şahin', 'baykuş', 'muhabbet']
```

Örnek 18’de extend komutu listelerdeki öğelerin kendi elemanlarını koruyarak genişletme işlemi yapmıştır.

5.4.6. 'indis' kullanımı

Örnek 19

```
sebzeler =["lahana","marul","pırasa","ıspanak","fasulye"]  
print(sebzeler.indis("ıspanak"))  
3
```

İndis metodu yardımıyla Örnek 19’da görüldüğü gibi verilen bir ögenin indis numarasını vermektedir.

5.4.7. 'clear' kullanımı

Örnek 20

```
liste=["ayva","nar","kiraz","kayısı","Üzüm"]
liste.clear()
print(liste)
[]
```

clear() metodu kullanılarak örnek 20'deki listenin tüm öğeleri silinmiştir.

5.4.8. 'pop' kullanımı

Örnek 21

```
sebzeler=["lahana","marul","pırasa","ıspanak","fasulye"]
sebzeler.pop(2)
print(sebzeler)
['lahana', 'marul', 'ıspanak', 'fasulye']
```

Örnek 21'de pop metodu ile sebzeler listesinden 2.indis numarasına ait olan **"pırasa"** adlı öge silinmiştir.

5.4.9. 'remove' kullanımı

Örnek 22

```
sehirler=["adana","ağrı","bursa","konya","ankara"]
sehirler.remove("konya")
print(sehirler)
['adana', 'ağrı', 'bursa', 'ankara']
```

Örnek 22'de öge adına göre silme işlemi yapılmıştır ve **"konya"** adlı öge sehirler listesinden silinmiştir.

MODÜL 5

5.4.10. 'reverse' kullanımı

Örnek 23

```
sayilar=[10,20,30,40,50,60,70]
sayilar.reverse()
print(sayilar)
[70, 60, 50, 40, 30, 20, 10]
```

Örnek 23'te reverse metodu ile liste öge elemanları tersten sıralanmıştır.

5.4.11. 'sort' kullanımı

Örnek 24

```
isimler=["elif","ayşe","kemal","kaan","hafsa"]
isimler.sort()
print(isimler)
['ayşe', 'elif', 'hafsa', 'kaan', 'kemal']
```

Sort metodu ile Örnek 24'te isimler listesi öğeleri alfabetik olarak sıralanmıştır.

5.4.12. 'del' kullanımı

Örnek 25

```
takimlar = ['GS', 'FB', 'BJK', 'TS']
del takimlar[2]
print(takimlar)
['GS', 'FB', 'TS']
```

Örnek 25'te del metodu ile takimlar listesine ait 'BJK' ögesi indis numarasına göre silinmiştir.

Yukarıdaki örneklerde temel liste metodlarının her birine yönelik örnek ve çıktıları verilmiştir. Ayrıca `__xxx__` şeklinde özel metodlar da bulunmaktadır. Bu metodlarda **dir(list)** şeklinde komut satırına yazıldığında aşağıdaki şekilde çıktı alınır.

```
[ '__add__', '__class__', '__contains__', '__delattr__', '__delitem__', '__dir__',
'__doc__', '__eq__', '__format__', '__ge__', '__getattr__', '__getitem__',
'__gt__', '__hash__', '__iadd__', '__imul__', '__init__', '__init_subclass__',
'__iter__', '__le__', '__len__', '__lt__', '__mul__', '__ne__', '__new__',
'__reduce__', '__reduce_ex__', '__repr__', '__reversed__', '__rmul__', '__setattr__',
'__setitem__', '__sizeof__', '__str__', '__subclasshook__', 'append', 'clear', 'copy',
'count', 'extend', 'index', 'insert', 'pop', 'remove', 'reverse', 'sort']
```

5.5. Len() Fonksiyonu ile Uzunluk Bilgisi

`len()` fonksiyonu, İngilizce `length`'in (uzunluk) kısaltılmış hâlidir. String ifadesinin uzunluğunu yani karakter sayısını verir. Örnek 14'te `a` adlı string değişkene değer atadığımızda değişkendeki karakterlerin sayısını vermektedir.

Örnek

26

```
a="Galatasaray"
print(len(a))
11
```

Örnek 26'da "`len()`" kullanımına bakıldığında, takımlar adında bir liste tanımlanmıştır. Bu listeye veri girişi yapılmıştır. `Len()` komutu ile listenin adı yazılarak, kaç elemanlı olduğu ekrana yazdırılmıştır. Örnek 27'de `len()` fonksiyonu ile takımlar listesinin eleman listesi 4 olarak verilmiştir.

MODÜL 5

Örnek

27

```
takimlar = ['GS', 'FB', 'BJK', 'TS']  
print( len(takimlar))  
4
```

Örnek 28’de 2 adet liste1 ve liste2 adında liste tanımlanmıştır. Bu listelere elemanlar girilerek, len() komutu ile kaç elemanlı olduğu bulunmuştur.

Örnek

28

```
liste1, liste2 = ['abc',56,74 , 'python'], [12, 'opencv','a']  
print ("İlk liste uzunlugu          : ", len(liste1))  
print( "İkinci listenin uzunluğu : ", len(liste2))  
İlk liste uzunlugu          : 4  
İkinci listenin uzunluğu    : 3
```

5.6. İç İç Listeler

Bir liste herhangi bir sıralama nesnesi içerebilir, hatta başka bir liste (alt liste) içerebilir, alt listeler de alt listeler içerebilir ve bu şekilde devam eder. Bu yuvalanmış liste olarak bilinir. Hiyerarşik yapılara veri düzenlemek için bunlar kullanılabilir.

Örnek

29

```
liste1 = [1,2,3]  
liste2 = [4,5,6]  
liste3 = [7,8,9]  
yeniliste = [liste1,liste2,liste3]  
print(yeniliste)  
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```


Örnek 29’da 3 adet liste oluşturuldu. Bu listelerin her biri ayrı öge olacak şekilde birleştirilerek iç içe liste oluşturuldu ve ekrana yazdırıldı.

Örnek 30’da sebzeler adında boş şekilde liste oluşturuldu.

Örnek**30**

```
sebzeler=[]
sebzeler.append(['yeşil','ıspanak'])
sebzeler.append(['beyaz','lahana'])
sebzeler.append(['turuncu','havuç'])
sebzeler.append(['siyah','turp'])
sebzeler.append(['kırmızı','domates'])
```

Append() metodunu kullanarak sebzeler adlı listeye 5 adet eleman girildi. Girilen verilerin listelenmesi:

```
print(sebzeler)
[['yeşil', 'ıspanak'], ['beyaz', 'lahana'], ['turuncu', 'havuç'], ['siyah', 'turp'],
['kırmızı', 'domates']]
```

Sebzeler adında listede 4 adet eleman bulunmaktadır. Bunları teker teker sıralanırsa:

```
print(sebzeler[0])
['yeşil', 'ıspanak']
```

Tablo 4. Sebzeler listesi elemanları

Liste adı	sebzeler				
Eleman değeri	‘yeşil’, ‘ıspanak’	‘beyaz’, ‘lahana’	‘turuncu’, ‘havuç’	‘siyah’, ‘turp’	‘kırmızı’, ‘domates’
İndis numarası	0	1	2	3	4

MODÜL 5

Örnek

31

```
sebzeler=[['yeşil','ıspanak'], ['beyaz','lahana'], ['turuncu','havuç']]
sebze=sebzeler[1]
print(sebze)
['beyaz', 'lahana']
```

Örnek 31'e bakıldığı zaman, sebze değişkenine sadece sebzeler [1] matrisindeki değer atanmıştır. Tablo 4'te sebzeler listesinde sadece havuç değeri listelenmek isteniyorsa indis numarasından faydalanılır.

Örnek

32

```
sebzeler=[['yeşil','ıspanak'], ['beyaz','lahana'], ['turuncu','havuç']]
print(sebzeler[2][1])
havuç
```

Örnek 33'te 4 adet liste oluşturulmuştur. İlk üç listenin tüm öğeleri son_liste adında listeye aktarılarak, ekrana yazdırılmıştır.

Örnek

33

```
# 3 Adet liste oluşturalım.
birinci_liste = [1,2,3]
ikinci_liste = ['a','b','c']
ucuncu_liste= [40,50,60]
son_liste= [birinci_liste,ikinci_liste,ucuncu_liste]
print(son_liste)
[[1, 2, 3], ['a', 'b', 'c'], [40, 50, 60]]
```

Örnek 34'te 3 adet liste oluşturulmuştur. Bunlara birinci_liste, ikinci_liste, ucuncu_liste isimleri verilmiştir. Son_liste adında liste oluşturularak diğer üç listenin elemanları bu listeye kaydedilmiştir. Ekrana sadece a ve 50 değerlerini yan yana listelemek için aşağıdaki kodlar kullanılır.

Örnek

34

```
# 3 Adet liste oluşturalım.
birinci_liste = [1,2,3]
ikinci_liste = ['a','b','c']
ucuncu_liste= [40,50,60]
son_liste= [birinci_liste,ikinci_liste,ucuncu_liste]
print(son_liste[1][0],son_liste[2][1])
a 50
```

5.7. Veri Tipi Dönüşümleri

Listelerde veri tipi dönüşümleri için, elemanlara yeni değer ataması yapıldığında string, int, float vb. veri tipleri arasında değer alabilir.

Örnek

35

```
liste=[1,2,3,4,5,'ankara']
print(liste)
liste[0]=str("kocaeli")
liste[2]=float(1.5)
liste[5]=int(20)
print(liste)
[1,2,3,4,5,'ankara']
['kocaeli', 2, 1.5, 4, 5, 20]
```

Örnek 35'te liste adında bir liste örneği tanımlanmıştır. İçindeki elemanları 1,2,3,4,5,'ankara' 'dır. İndis numarasına göre liste [0] değeri önce 1 iken veri tipi dönüşümünden dolayı 'kocaeli' olmuştur. Liste [2]'in değeri ise ilk başta 3'tür. Daha sonra float veri tipinde 1.5 değerini almıştır. Son olarak liste[5]'in değeri 'ankara'dır. Liste[5]'e int veri tipinde bir değer kaydedilerek 20 olmuştur.

Herhangi bir string türünde veriyi parçalayarak da liste oluşturulabilir. Örnek 36'da string veri türünde adı meyve olan bir değişken olarak tanımlanmış ve veri olarak elmayı atanmıştır. Meyve adlı değişken liste yardımıyla parçalanmıştır.

MODÜL 5

Örnek

36

```
meyve="elma"  
liste=list(meyve)  
print (liste)  
['e', 'l', 'm', 'a']
```

Örnek 37’de 1 ile 15 arasındaki sayılardan oluşan liste oluşturulmuştur. Listenin öğeleri ekrana listelenmiştir. Ekrana listeleme işleminde `sort()` metodu ile öğeler küçükten büyüğe, `reverse` ile büyükten küçüğe doğru sıralanmıştır.

Örnek

37

```
liste=list(range(1,15,2))  
print(liste)  
liste.sort()  
print(liste)  
liste.reverse()  
print(liste)  
[1, 3, 5, 7, 9, 11, 13]  
[1, 3, 5, 7, 9, 11, 13]  
[13, 11, 9, 7, 5, 3, 1]
```

split() metodu, listeyi belirtilen ayırıcı kullanarak yeniden döndürmeye yarar. Yani `split()` karakter dizilerini istenen şekilde böler. -ayırıcı diye tanımladığımız ilk parametre, karakter dizisinin nereden bölüneceğini seçer. Eğer ayırıcı tanımlanmazsa karakter dizisi her boşluk gördüğünde ayırır. Örnek 38’de bilgiler girildikçe listeye kaydedecektir. Listeye 4 adet öge girilmiş ve listeleme işlemi yapılmıştır.

Örnek

38

```
bilgi=input("bilgilerinizi araya virgül koyarak yazınız: ")
liste=bilgi.split(",")
print(liste)
bilgilerinizi araya virgül koyarak yazınız: Hafsa,Meva,Konya,1
['Hafsa', 'Meva', 'Konya', '1']
```

Örnek 39’da ise cümle değişkenindeki kelimeler split metodu ile kelimeler listesine aktarılmıştır. Len() metodu ile de kaç adet öge olduğu ekrana listelenmiştir.

Örnek

39

```
cumle="23 nisan herkese mutlu olsun"
kelimeler=cumle.split(" ")
print("cümleinizde ",len(kelimeler),"adet kelime vardır")
cümleinizde 5 adet kelime vardır
```

5.8. Bölüm Sonu Örnekleri

1. Harfler adıyla bir liste oluşturup içine 'a', 'e', 'i', 'o', 'ı', 'u' elemanları kaydediniz. Bu listede i ve p harflerinin sayısını ekrana yazdırınız.
2. Liste1, liste2, liste3 ve liste4 adında dört adet liste oluşturup aynı satırda olacak şekilde tanımlayıp, her bir listeye birer adet eleman girip listeleyiniz.
3. İki adet liste tanımlayarak bu iki listeyi "+" operatörü ile toplama işlemi yaptırılıp, üçüncü bir listeye atama işlemi yapınız. Son listeyi ekrana yazdırınız.
4. Aşağıdaki listeyi hem küçükten büyüğe hem de büyükten küçüğe doğru sıralanacak şekilde ekrana listeleyiniz.

```
liste = [34,1,56,334,23,2,3,19]
```

5. Aşağıdaki kodları verilen liste örneğinde ekran çıktısını yazınız.

```
listem= ["Merhaba", "Türkiye", "Nasılsın", "Tebrیکler"]  
print(listem[-1])  
print(listem[-3])  
print(listem[-4])
```

6. Aşağıdaki örnekte liste tanımlanmış ve ekran çıktısı verilmiştir. Ekran çıktısının aşağıdaki gibi olması için boş kalan kısımları tamamlayınız.

```
liste = [1, 2, 3, 4, 5, 6, 7]
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
[2, 3]
```

```
[1, 2, 3]
```

```
[4, 5, 6, 7]
```

```
[1, 2, 3, 4, 5, 6, 7]
```

7. Aşağıda verilen örnekte boş kalan kısımları doldurunuz.

```
isimler = ['ali', 'veli', 'ayşe']
```

```
.....
```

```
ad_soy1 = isimler[0] + ' ' + soyisimler[0]
```

```
.....
```

```
ad_soy3 = isimler[2] + ' ' + soyisimler[2]
```

```
print(ad_soy1)
```

```
.....
```

```
print(ad_soy3)
```

```
ali türk
```

```
veli izci
```

```
ayşe erel
```

8. liste = ['bir', 'iki', 'dört']

```
print(liste)
```

['bir', 'iki', 'dört'] şeklinde ekran çıktısı olmaktadır. Ama ['bir', 'iki', 'üç', 'dört', 'beş'] olacak şekilde listeyi güncelleyiniz.

9. Aşağıdaki listede listenin ilk ve son verilerine ulaşmak ve listelemek için gerekli kodları yazınız.

```
liste=["birinci veri", "ikinci veri", "üçüncü veri ", "dördüncü veri",  
"beşinci veri"]
```

Cevaplar

1. Aşağıdaki örnekte harfler adında bir liste tanımlanmış ve 6 adet eleman girilmiştir. count metodu kullanılarak, i ve p harflerinin sayısı ekrana yazdırılmıştır. i harfi için 2 ve p harfi için 0 değeri listelenmiştir.

```
harfler = ['a', 'e', 'i', 'o', 'i', 'u']
count = harfler.count('i')
print('i harflerinin sayısı:', count)
count = harfler.count('p')
print('p harflerinin sayısı:', count)
i harflerinin sayısı: 2
p harflerinin sayısı: 0
```

2. Aşağıdaki örnekte 4 adet liste tanımlanmıştır. Bu listelere elemanlar atanarak listelenmiştir.

```
liste1,liste2,liste3,liste4= ['a',100,3.14,'python']
print(liste1)
print(liste2)
print(liste3)
print(liste4)
a
100
3.14
python
```

3.

```
list1 = ["a", "b" , "c"]
list2 = [1, 2, 3]
list3 = list1 + list2
print(list3)
['a', 'b', 'c', 1, 2, 3]
```


4. `liste = [34,1,56,334,23,2,3,19]`
`liste.sort()`
`print('küçükten büyüğe doğru',liste)`
`liste.reverse()`
`print('büyükten küçüğe doğru',liste)`
 küçükten büyüğe doğru `[1, 2, 3, 19, 23, 34, 56, 334]`
 büyükten küçüğe doğru `[334, 56, 34, 23, 19, 3, 2, 1]`
5. `listem= ["Merhaba", "Türkiye", "Nasılsın", "Tebrikler"]`
`print(listem[-1])`
`print(listem[-3])`
`print(listem[-4])`
 Tebrikler
 Türkiye
 Merhaba
6. `liste = [1, 2, 3, 4, 5, 6, 7]`
`print(liste[1:3])`
`print(liste[:3])`
`print(liste[3:])`
`print(liste[:])`
`[2, 3]`
`[1, 2, 3]`
`[4, 5, 6, 7]`
`[1, 2, 3, 4, 5, 6, 7]`

MODÜL 5

7.

```
isimler = ['ali','veli','ayşe']
soyisimler = ['türk','izci','erel']
ad_soy1 = isimler[0] +' ' + soyisimler[0]
ad_soy2 = isimler[1] +' ' + soyisimler[1]
ad_soy3 = isimler[2] +' ' + soyisimler[2]
print(ad_soy1)
print(ad_soy2)
print(ad_soy3)
ali türk
veli izci
ayşe erel
```
8.

```
liste = ['bir','iki','dört']
liste[2]='üç'
liste.insert(3,'dört')
liste.insert(4,'beş')
print(liste)
['bir', 'iki', 'üç', 'dört', 'beş']
```
9.

```
liste=["birinci veri","ikinci veri","üçüncü veri ","dördüncü veri","beşinci veri"]
#beş elemanlı listenin ilk verisi
print(liste[0])
#beş elemanlı listenin son verisi
print(liste[4])
birinci veri
beşinci veri
```